DOI 10.2478/ama-2022-0007

FUZZY SYNERGETIC CONTROL FOR DYNAMIC CAR-LIKE MOBILE ROBOT

Zoulikha BOUHAMATOU*, **Foudil ABEDSSEMED***

*Department of Electronics, Faculty of Technology, University Mostefa Ben Boulaïd, Batna2, Algeria

z.bouhamatou@univ-batna2.dz, f.abdessemed@univ-batna2.dz

received 25 May 2021, revised 27 September 2021, accepted 17 October 2021

Abstract: This paper aims to present the dynamic control of a Car-like Mobile Robot (CLMR) using Synergetic Control (SC). The SC control is used to make the linear velocity and steering velocity converge to references. Lyapunov synthesis is adopted to assure controlled system stability. To find the optimised parameters of the SC, the grey wolf optimiser (GWO) algorithm is used. These parameters depend on the best-selected fitness function. Four fitness functions are selected for this purpose, which is based on the integral of the error square (ISE), the integral of the square of the time-weighted error (ITSE), the integral of the error absolute (IAE) and the integral of the absolute of the time-weighted error (TIAE) criterion. To go further in the investigation, fuzzy logic type 2 is used to get at each iteration the appropriate controller parameters that give the best performances and robustness. Simulations results are conducted to show the feasibility and efficiency of the proposed control methods.

Key words: CLMR, Synergetic Control, Lyapunov stability, GWO, Fuzzy Logic type2

1. INTRODUCTION

In recent years, many works have been done on self-driving cars. This is mainly due to the increasing number of accidents on the road caused especially by inattention. How we drive is inefficient and becomes a real problem as we get old or we get more infirm. The way technology is evolving has brought many solutions to rolling vehicles. To end up with a smart vehicle, we should equip it with different sensors, a robust computing platform and robust control algorithms that execute in real time. In fact, the research in control focuses on the robustness of control laws and their influence on external disturbance, as well as on vehicle's response speed to execute the task and perform the desired objectives. Therefore, the recent works on control are oriented to these kinds of systems. The controller is to regulate some of the states of the vehicle such as velocity and rotation rate by sensing the current state variables and generating actuator signals to satisfy the objectives provided. Within this aspect, different techniques have been proposed. To obtain a stable movement of trajectory tracking we can find a lot of works applied to indoor and outdoor mobile robots. Dung et al. [4] developed an adaptive sliding mode control, they use this control to achieve the mobile robot tracking a smooth curved reference, with a desired constant velocity. Yeh et al. [20] presented an adaptive fuzzy sliding-mode control for trajectory tracking under disturbances for a car-like mobile robot (CLMR). They presented three dynamic controllers, namely Adaptative Sliding Mode Control(ASMDC), and Adaptative Fuzzy Sliding Mode Control (AFSMDC), which are proposed to reduce the effect of the problem of uncertainties and external disturbances. Peng and Shi [16] solved the problem of uncertainties and external disturbances in a non-holonomic wheeled mobile robot, where the uncertainties are approximated by a fuzzy logic system and an adaptive fuzzy integral terminal sliding mode controller is used for velocity controller. An adequate control law is presented in Benaziza et al. [2]. It is based on Global Terminal Sliding Mode (GTSM) with fuzzy control. The objective of this control is to eliminate the disturbances of the angular and linear velocities, respectively. Moreover, an exponential reaching law is presented in a dynamic model to eliminate the uncertainties. Ibrahim [8] proposed a robust sliding mode controller for trajectory tracking for the non-holonomic robot. In his study, two stages of the proposed control strategy are presented. The first one uses the steering controller for the kinematics model and the second is a robust sliding mode control technique for the velocity. Mallem et al. [12] proposed a dynamic control RBF Global Fast Sliding mode for a mobile robot, for which the main task is making the linear and angular velocities converge to references in finite time. The system injected by disturbances. To stabilise the velocity errors and estimate the non-linear function of the robot, the RBF-GFSM approach is used, which combines the RBF neural network and Global Fast Sliding mode.

Sliding Mode Control (SMC) has been widely used in mobile robot control and great effort has been made to reduce its main inconvenience: chattering. Many approaches have been proposed to eliminate chatter, including synergistic control of the system and decoupling, but it is used for several advantages. First, it is well suited for digital implementation. The second advantage is that the control operates at the constant switching frequency and therefore avoids chattering problems.

As there is not much application of this type of control in the robot, Liu and Hsiao [11] proposed a finite time synergetic control (FTSC) for controlling robot manipulators. This technique was extracted from the use of synergetic theory and a terminal attractor technique. Their control scheme demonstrates its advantage in the finite-time convergence and chattering-free phenomena. Podvalny and Vasiljev [17] deployed a SC to solve the problem of synthesis of a multirotor unmanned aerial vehicle (UAV) regulator.

Veselov et al. [19] solved the problem of a group of mobile robots using the principles and methods of synergistic control theory. Sklyarov et al. [18] presented the explanation of using new non-linear approaches to design control systems for omnidirectional mobile robots, the synergetic control theory (SCT). Bhattacharyya et al. [3] proposed a brain-machine interfacing (BMI) paradigm for controlling the direction of end-point movement of a 3-degrees of freedom (DOF) robot arm, and they use a synergetic algorithm to manage a peripheral redundancy in multi-DOF joints towards energy optimality. Humaidi et al. [7] developed a SCT for controlling a one-link robot arm actuated by pneumatic artificial muscles (PAMs) in opposing bicep/tricep positions. A terminal SC is applied to a nonlinear helicopter model to control the nonlinear fifth-order helicopter model, which controls height and angle [6].

Moreover, when it comes to looking for the best performances that the system can achieve, some parameters should be well determined according to certain predefined criteria. This can always be obtained by intelligent optimisation algorithms. One of these algorithms is grey wolf optimiser (GWO). It is an evolutionary algorithm that uses a population of candidate solutions to develop an optimal solution to the problem. Mirjalili et al. [14] proposed this algorithm in 2014. It was originally inspired by the living world, more precisely by the social behaviour of animals evolving in swarms, inspired by grey wolves (Canis lupus). The GWO algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. GWO has gained increasing popularity among researchers and practitioners as a robust and effective technique for solving difficult optimisation problems [5, 6, 9, 13, 15].

In this paper, we present a SC law to command successfully a CLMR. In fact, it seems reasonable to work on controlling autonomous driving vehicles by taking a CLMR. This is accom-plished by making the velocity and the rotational rate of the car-like mobile robot converge to their references in presence of disturbances. The asymptotic stability is guaranteed by Lyapunov theory, for which certain parameters need to be well chosen to achieve the best performances. To obtain the optimised parame-ters, the grey wolf optimiser (GWO) algorithm is employed for this purpose.

The paper is organised as follows, in section two, the kinematic and dynamic model of the CLMR is introduced. Section 3 introduces the basics of SCT for nonlinear systems. It is reviewed briefly with stability proved for closed-loop control systems. Section 4, GWO is applied to determine the optimal parameters of SC. Section 5, propose a Fuzzy Logic type2 Synergetic Control for an uncertain CLMR system with external disturbances. In Section 6, simulations are performed to examine the feasibility and effectiveness of the approach. A conclusion is drawn in Section 7.

2. KINEMATICS AND DYNAMICS MODELS

This section aims to describe the kinematic and dynamic model of a CLMR. The geometry of the CLMR is shown in Fig. 1.

2.1. Kinematics model

The state of the robot's motion is represented by the vector q such that: $q = [x \quad y \quad \theta \quad \psi \quad \varphi]^T$, where (x, y) are the coordinates of the center of the two rear wheels. θ: indicates the heading direction of the CLMR concerning the x-axis, ψ is the steering angle of the front wheels, and φ denotes the angular velocity of the wheels. The kinematic model of the CLMR is given by,

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} cos(\theta) & 0 \\ sin(\theta) & 0 \\ tan(\psi)/d & 0 \\ 0 & 1 \\ \frac{1}{a} & 0 \end{bmatrix} \begin{bmatrix} v \\ \dot{\psi} \end{bmatrix} = S(q)V \tag{1}$$

where: $S(q)\epsilon\ \mathbb{R}^{5x2}$and $V\epsilon\ \mathbb{R}^2$ are the full rank velocity transformation matrix and velocity vector, respectively. $\dot{\psi}\epsilon\mathbb{R}$ denotes the steering velocity of the front wheels, d is the distance between the front and rear wheels, and a is the radius of the wheels.

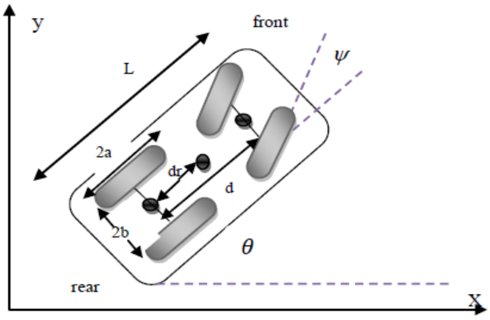

Fig. 1. The kinematics model of a non-holonomic CLMR. CLMR, Car-like mobile robot

The linear velocity of the CLMR, *v(t)*$\epsilon\ \mathbb{R}$, is obtained as:

$$v(t) = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{2}$$

Here, we have two non-holonomic constraints, one for each wheel pair, that is:

$$\dot{x} sin\theta - \dot{y}\, cos\, \theta = 0 \tag{3}$$

$$\dot{x}\, sin(\theta + \varphi) - \dot{y}\, cos(\theta + \varphi) - \rho\dot{\theta}\, cos\, \varphi = 0 \tag{4}$$

The two constraints can be rewritten as $H(q)\dot{q} = 0$, and therefore,

$$H(q)S(q)V = 0 \tag{5}$$

Such that

$$H(q) \begin{bmatrix} sin\,\theta & -cos\,\theta & 0 & 0 & 0 \\ sin(\theta + \varphi) & -cos(\theta + \varphi) & \rho\, cos\, \varphi & 0 & 0 \end{bmatrix} \tag{6}$$

2.2. The dynamic model of the CLMR

The robot dynamic model of the non-holonomic mobile robot deals with the derivation of the dynamic equations of the robot motion. In this case, we use the methodologies of the Lagrange formula,

$$\frac{\mathrm{d}}{\mathrm{dt}}\left(\frac{\partial \mathrm{L}}{\partial \dot{\mathrm{q}}_{\mathrm{i}}}\right) - \frac{\partial \mathrm{L}}{\partial \mathrm{q}_{\mathrm{i}}} = \overline{\mathrm{N}}(\mathrm{q})\tau + \mathrm{H}^{\mathrm{T}}(\mathrm{q})\lambda \tag{7}$$

L is the Lagrangian function defined by:

$$L = K\text{-}P \tag{8}$$

Here, *K* is the total kinetic energy, and *P* is the total potential energy of the robot. The Lagrangian *L* is equal to *K* since the robot is moving on a horizontal plane and so the potential energy *P* is zero.

$$\frac{d}{dt}\left(\frac{\partial K}{\partial \dot{q}_i}\right)-\frac{\partial K}{\partial q_i}=\bar{N}(q)\tau+H^T(q)\lambda \tag{9}$$

Where *i = 1,…,5* and λ is a Lagrange multiplier associated with the constraints, τ is the torque input vector. The kinetic energy of CLMR can be described as:

$$K = K_p + K_{fw} + K_{rw} \tag{10}$$

where K_p, K_{fw}, K_{rw} denote the kinetic energy of the body, front wheels and rear wheels of CLMR. According to Lagrange mechanics, the following matrices are adopted for CLMR models [2].

$$\bar{M}(q)\ddot{q}+\bar{V}_m(q,\dot{q})\dot{q}=\bar{N}(q)\tau+H^T\lambda \tag{11}$$

where: $\bar{M}$(q)$\epsilon\mathbb{R}^{5x5}$is a symmetric positive definite inertia trix,$\bar{V}_m(\mathrm{q},\dot{\mathrm{q}})\epsilon\mathbb{R}^{5x5}$ is the centripetal and Coriolis matrix, $\bar{N}$(q)$\epsilon\mathbb{R}^{5x2}$ is the input transformation matrix, H(q)$\epsilon\mathbb{R}^{5x2}$ is a matrix associated with the nonholonomic constraints, and $\dot{q}$ and $\ddot{q}$denote the velocity and acceleration vectors, respectively. The matrices in (11) are found to be,

$$\bar{\mathrm{M}}(\mathrm{q}) = \begin{bmatrix} \mathrm{m} & 0 & -\mathrm{msin}\theta & 0 & 0 \\ 0 & \mathrm{m} & \mathrm{mcos}\theta & 0 & 0 \\ -\mathrm{msin}\theta & \mathrm{mcos}\theta & \mathrm{I}_\theta & 2\mathrm{I}_\mathrm{w} & 0 \\ 0 & 0 & 2\mathrm{I}_\mathrm{w} & 2\mathrm{I}_\mathrm{w} & 0 \\ 0 & 0 & 0 & 0 & 8\mathrm{I}_\mathrm{w} \end{bmatrix}$$

$$\bar{V}_m(q,\dot{q}) = \begin{bmatrix} 0 & 0 & -m\dot{\theta}cos\theta & & 0 & 0 \\ 0 & 0 & -m\dot{\theta}sin\theta & & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\bar{N}(q) = \begin{bmatrix} cos\theta & sin\theta & dsin\psi cos\psi & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^T$$

The general dynamic model of mobile robot with unknown disturbances can be described by the following equation:

$$\bar{M}(q)\ddot{q}+\bar{V}_m(q,\dot{q})\dot{q} = \bar{N}(\mathrm{q})\tau+\tau_d+H^T\lambda \tag{12}$$

where τ_d is a denoted bounded unknown disturbance including unstructured but not modelled dynamically. It would be more suitable to express the dynamic equations of motion in terms of internal velocities. Substituting (1) and its differentiation in (12) and pre-multiplying by $S^T(q)$, results in Equation (13).

$$M(q)\dot{V}+V_m(q,\dot{q})V = \mathrm{N(q)}\tau+\tau_{vd} \tag{13}$$

Where, $M = S^T\bar{M}S\epsilon\mathbb{R}^{2x2}$, $V_m = S^T[\,\bar{M}\dot{S}+\bar{V}_mS]\epsilon\mathbb{R}^{2x2}$, N(q) =$S^T\bar{N}\epsilon\mathbb{R}^{2x2}$;$\tau_{vd} = S^T * \tau_d\epsilon\mathbb{R}^{2x2}$.

The dynamic Equation (13) of the non-holonomic CLMR can be rewritten as:

$$\dot{V}(t) = -AV(t)+B\tau(t)+d \tag{14}$$

where, $A = M^{-1}V_m$, $B = M^{-1}N$, $d = M^{-1}\tau_{vd}$. In case of no disturbances ($d = 0$), Equation (14) reduces to

$$\dot{V}(t) = -AV(t)+B\tau(t) \tag{15}$$

3. SC DESIGN OF THE CAR LIKE MOBILE ROBOT

In this section, the SC method is used in designing a dynamic tracking controller. The objective is to control the velocity and rotation rate to track the desired references.

SC system is an invariant-manifold-based control method and can be applied for controlling nonlinear dynamic systems. In the sequel, we present the basics of SC synthesis for a nonlinear dynamic system described by:

$$\dot{v} = f(v,u,t) \tag{16}$$

where *v* is the state vector, *u* is the control input vector, and *t* is time. The design process of SC algorithm for a nonlinear dynamic system can be summarised in the following steps as follows:

- Define the macro variableσ (v, t), as a function of the system state. This macro variable will be used to determine a stabilising control law.

$$u(v) = u(v,\sigma(v)) \tag{17}$$

The system is forced by the controller to operate on the manifold: $\sigma = 0$.

- Design a control law that would drive the system states onto the specified manifold and remain on it with an evolution constraint, which can be stated as a set of the dynamic evolution of macro-variables written as Equation (18):

$$T\dot{\sigma}(v,t)+\sigma(v,t) = 0 \tag{18}$$

where $T = [T1, T2]T$ is the rate of convergence vector of the macro-variables to manifolds $\sigma = 0$, and $\dot{\sigma}$is the derivative of the aggregated macro variable by time.

- Solve the system (17) with the evolution condition (18) to obtain the control law.

To solve the differential Equation (18), we select a PI-type of the macro variable for v_c and ψ_c. By doing so, the tracking errors would converge effectively. Hence, the macro variables defined are expressed as:

$$\sigma(\boldsymbol{v},t) = \begin{bmatrix} \sigma_v\,(t) \\ \sigma_\psi(t) \end{bmatrix} = \begin{bmatrix} k_1e_v+\int_0^t e_v\,(t)dt \\ k_2e_\psi+\int_0^t e_\psi\,(t)dt \end{bmatrix} \tag{19}$$

The derivation of the macro variable$\sigma(t)$ is:

$$\dot{\sigma}(\boldsymbol{v},t) = \begin{bmatrix} \dot{\sigma}_v\,(t) \\ \dot{\sigma}_\psi(t) \end{bmatrix} = \begin{bmatrix} k_1\dot{e}_v+e_v \\ k_2\dot{e_\psi}+e_\psi \end{bmatrix} \tag{20}$$

In a more compact form, we write:

$$\dot{\sigma}(\boldsymbol{v},t) = k\dot{e}+e \tag{21}$$

where$k = [k_1, k_2]^T$ is the vector of positive coefficients and the vector of errors is $e = [e_v \quad e_\psi]^T$, e_v is the linear velocity error and e_ψ is the steering velocity error. It is obvious that the tracking errors converge to if the parameters (k_1, k_2) are selected properly. The control law τ is obtained by letting $\dot{\sigma}(t)$ equal to zero. This is necessary for the state trajectory to stay on the synergetic surface. Since $T\dot{\sigma}+\sigma =$0, therefore:

$$\dot{\sigma} = \frac{-\sigma}{T} \tag{22}$$

Substituting (22) in (21), yields

$$\dot{e} = -\frac{e}{k}-\frac{\sigma}{T} \tag{23}$$

The velocities errors are defined as:

$$e = V_d - V \tag{24}$$

For which the derivative is obtained:

$$\dot{e} = \dot{V}_d - \dot{V} \tag{25}$$

According to Equation (15), the system dynamic error is obtained as :

$$\dot{e} = \dot{V}_d(t) + AV(t) - B\tau(t) \tag{26}$$

Upon solving Eq. (26) for τwhile taking into account Equation (23), the SC for the car like mobile robot law can be found as:

$$\tau(t) = B^{-1}[\frac{e}{k} + \frac{\sigma}{kT} + \dot{V}_d(t) + AV(t)] \tag{27}$$

where k is a diagonal matrix whose elements are k_1 and k_2.By a suitable selection of the design parameters k_1 and k_2, the final system can attain a suitable performance. Stability can be evaluated using the following Lyapunov function candidate

$$L_1 = \frac{1}{2}\sigma^T\sigma \tag{28}$$

This leads, after differentiation and using Eq. (21), to:

$$\dot{L}_1 = \sigma^T(k\dot{e} + e) \tag{29}$$

Written finally as,

$$\dot{L}_1 = -\frac{\sigma^T\sigma}{T} < 0 \tag{30}$$

Therefore, $\dot{L}_1$ is confirmed negative and consequently, the velocity tracking error will exponentially converge to zero.

4. GWO OF PARAMETERS

Grey Wolf Optimisation (GWO) is a smart swarm technique developed by Mirjalili et al. (2014), which mimics the leadership hierarchy of wolves which are well known for their group hunting. This algorithm mimics the social leadership hunting behavior of gray wolves in the wild. In this algorithm, the population is divided into four groups: alpha (α), beta (β), delta (δ) and omega (ω). The first three strongest wolves are considered α, β and δ which guide the other wolves (ω) to promising areas of the search space.

4.1. 4.1. Objective function

The objective function should be well chosen in order to find the unknown parameters of the designed control law leading to improve the performances of the control loop, such as the dynamic precision, the overshoot and the static error. The objective function considered is based on an error criterion, which is often the case when evaluating controller performances, for our applications, we have carried out a series of tests. In the end, our choice fell on two criteria namely: The integral of the error square (ISE), the integral of the square of the time-weighted error (ITSE), the integral of the error absolute (IAE) and the integral of the absolute of the time-weighted error (TIAE):

$$ISE=\int_0^t e(t)^2 dt. \tag{31}$$

$$ITSE=\int_0^t te(t)^2 dt \tag{32}$$

$$IAE=\int_0^t |e(t)| dt \tag{33}$$

$$ITAE=\int_0^t t|e(t)| dt \tag{34}$$

4.2. GWO basic algorithm

This algorithm imitates the hunting mechanism of gray wolves in nature, this kind of wolves mostly prefer to live in groups. Where the group can consist of 5–12 wolves. Or in the GWO strategy, each individual in the group has a specific role. Moreover, one of the interesting realities of the social life of these wolves is a very strict social hierarchy structure in the group; the main stages of the hunting of this type of wolves are:

- Tracking, chasing and approaching the prey
- Pursuing, encircling and harassing the prey until it stops moving
- Attack towards the prey

When designing GWO the social hierarchy of wolves, we consider the alpha (α) the fittest solution, the second and third-best solutions are beta (β) and delta (δ), respectively. The rest of the candidate solutions are assumed to be omega. In the GWO algorithm, the hunting (optimisation) is guided by α, β and δ. The (ω) wolves follow these three wolves.

4.2.1. Encircling prey

Describe the grey wolves encircle prey during the hunt by the following equations:

$$\vec{D} = |\vec{C}.\vec{X}_p(t) - \vec{X}(t)| \tag{35}$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A}.\vec{D} \tag{36}$$

where: *t* indicates the current iteration, $\vec{A}$and $\vec{C}$ are coefficient vectors,$\vec{X}_\mathrm{p}$ is the position vector of the prey,$\vec{X}$ indicates the position vector of a grey wolf. The vectors are calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r_1} - \vec{a} \tag{37}$$

$$\vec{C} = 2\vec{r_2} \tag{38}$$

where components of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iterations and r_1 , r_2 are random vectors in [0,1].

4.2.2. Hunting

To search and identify the location of the prey and its surroundings. It is generally via Alpha, Beta and Delta which can also participate from time to time. However, the ideal location of prey does not have any idea of it remains. To mathematically simulate the hunting behavior of grey wolves, we assume that alpha (best candidate solution) beta and delta have better knowledge of the likely location of prey. Therefore, we save the three best solutions obtained so far and force other agents (including omegas) to update their sites according to the position of the best search agents.

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1}.\vec{X}_\alpha(t) - \vec{X}(t)|, \overrightarrow{D_\beta} = |\overrightarrow{C_2}.\vec{X}_\beta(t) - \vec{X}(t)|,$$
$$\overrightarrow{D_\delta} = |\overrightarrow{C_3}.\vec{X}_\delta(t) - \vec{X}(t)| \tag{39}$$

$$\overrightarrow{X_1} = \vec{X}_\alpha - \overrightarrow{A_1}.\overrightarrow{(D_\alpha)}, \overrightarrow{X_2} = \vec{X}_\beta - \overrightarrow{A_2}.\overrightarrow{(D_\beta)},$$
$$\overrightarrow{X_3} = \vec{X}_\delta - \overrightarrow{A_3}.\overrightarrow{(D_\delta)} \tag{40}$$

DOI 10.2478/ama-2022-0007

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \tag{41}$$

where: $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ represents the position of the alpha, beta and delta respectively. *X* indicates the position of the current tion. $\overrightarrow{C_1}, \overrightarrow{C_2}, \overrightarrow{C_3}$: Are random vectors.

4.2.3. Attacking prey (exploitation)

The proceeding of the hunt is finished when the prey stops moving which is then attacked by the grey wolves. In the mathematical model, the approaching towards the prey is made by a decreasein the value of a. Note that the fluctuation range of $\vec{A}$ is also decreased by a. $\vec{A}$ is a random value in the interval [-a,a], and for each iteration, the value of a is decreased from 2 to 0. When random values of $\vec{A}$ is in the interval [–1, 1], the next search agent position is between its current position and the position of the prey. For *A* value of |*A*|<*1* forces the wolves to attack towards the prey.

4.3. Algorithm process

The GWO algorithm works according to the following steps:

- Step 1: Initialise a random wolf population based on the upper and lower limits of the variables.
- Step 2: Calculate the corresponding objective value for each Wolf.
- Step 3: Choose the first three best wolves and save them under α, β and δ.
- Step 4: Update the position of the rest of the population (wolves) using Equations (39), (40), and (41).
- Step 5: Update of parameters a, A and C.
- Step 6: Go to the 2nd step if the final criterion is not satisfied.
- Step 7: Returns the position of α as the approximate optimal optimum.

Fig. 2 resumes the control strategy proposed in this work.

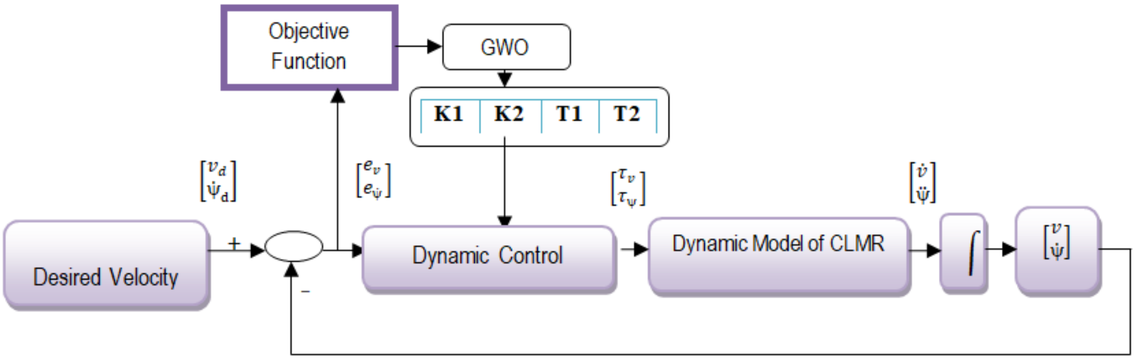
Objective Function
GWO
K1
K2
T1
T2
$\begin{bmatrix} v_d \\ \dot{\psi}_d \end{bmatrix}$
$\begin{bmatrix} e_v \\ e_{\dot{\psi}} \end{bmatrix}$
$\begin{bmatrix} \tau_v \\ \tau_\psi \end{bmatrix}$
$\begin{bmatrix} \dot{v} \\ \ddot{\psi} \end{bmatrix}$
Desired Velocity
+
–
Dynamic Control
Dynamic Model of CLMR
$\int$
$\begin{bmatrix} v \\ \dot{\psi} \end{bmatrix}$

Fig. 2. The system architecture of the closed-loop system

5. FUZZY LOGIC TYPE 2 PARAMETERS FINDING

Even though the GWO algorithm gives the parameters that are relatively suitable for the controller and can be viewed as a means to find the best choice than using trial and error, we remarked that the tracking is not as good as it was expected when disturbances are present. In this section, we use fuzzy logic type 2 to estimate the parameters$[T_1, T_2, k_1, k_2]^T$at each sampling time to reject as much as possible the effect of the external disturbances. In the presence of disturbances, the term d is added and Equation (14) presents, in this case, the system with added disturbances:

$$\dot{V}(t) = -AV(t) + B\tau(t) + d \tag{42}$$

In this equation the vector of disturbances is d, such that:$d = [dv\ d\dot{\psi}]$.The elements of the this vector are bounded, i.e., $|dv| < \zeta_v, |d\dot{\psi}| < \zeta_{\dot{\psi}}$. dv and $d\dot{\psi}$ represent the perturbations of linear and steering Velocities and ζ_v and $\zeta_{\dot{\psi}}$ are positive constants. In our study, we define dv and $d\dot{\psi}$ as:

$$\begin{cases} dv = sint(\pi t) \\ d\dot{\psi} = cost(\pi t) \end{cases} \tag{43}$$

The fuzzifier is built using three Gaussian membership functions for both input variables and three Gaussian membership functions for each of the four output variables as is depicted in Figs. 3–6. The following linguistic variables are assigned to each of the input and output fuzzy membership functions *N*, *Z* and of output *S*, *M* and *B*. Where the meaning of each linguistic variable should be clear from its mnemonic; in fact, *N* stands for negative, *Z* stands for Zero, *P* stands for positive, *S* stands for small, *M* stands for Mean and *B* stands for Big.

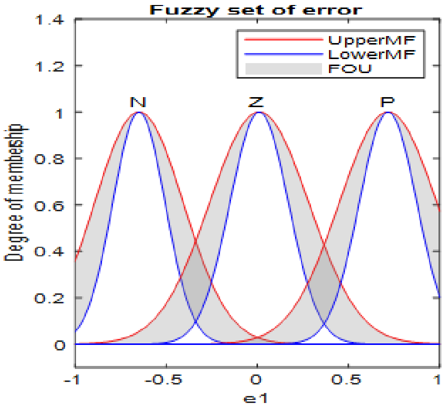

Fig. 3. Fuzzy sets of input function (e_v)

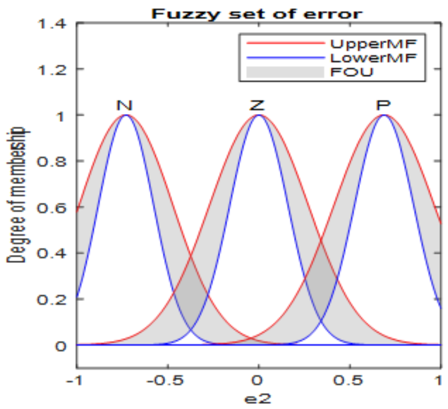

Fig. 4. Fuzzy sets of input function (e_{ψ})

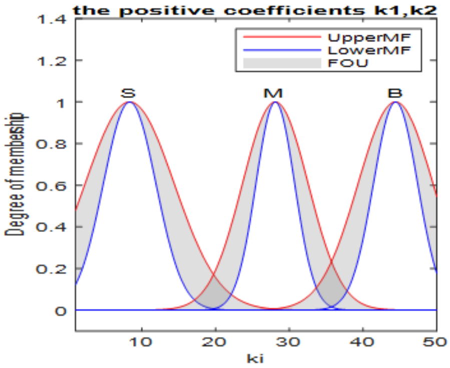

Fig. 5. Fuzzy sets of output function (K_i)

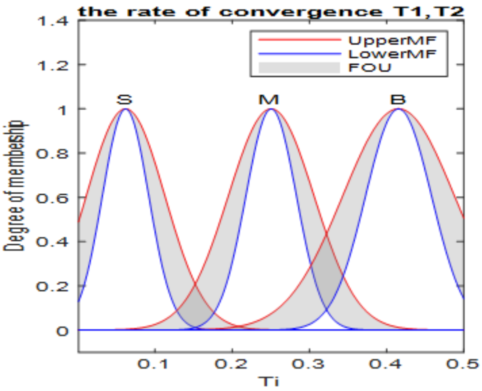

Fig. 6. Fuzzy sets of output function (T_i)

Fig. 7 shows the block diagram used for this purpose. In this system, the fuzzy inputs are the linear and steering velocity errors $\mathrm{e_v}$ and $\mathrm{e_\psi}$ while the fuzzy outputs are the SC parameters $\mathrm{k_1, k_2, T_1}$ and $\mathrm{T_2}$.

The derivation of the rules obeys many techniques such as that of the experience and skilled operator. In our case, we made use of the knowledge of the behavior of the systems towards the values of the different parameters, taking into consideration that $k_i \geq 0$ and $T_i \geq 0$, *i* = *1, 2*.

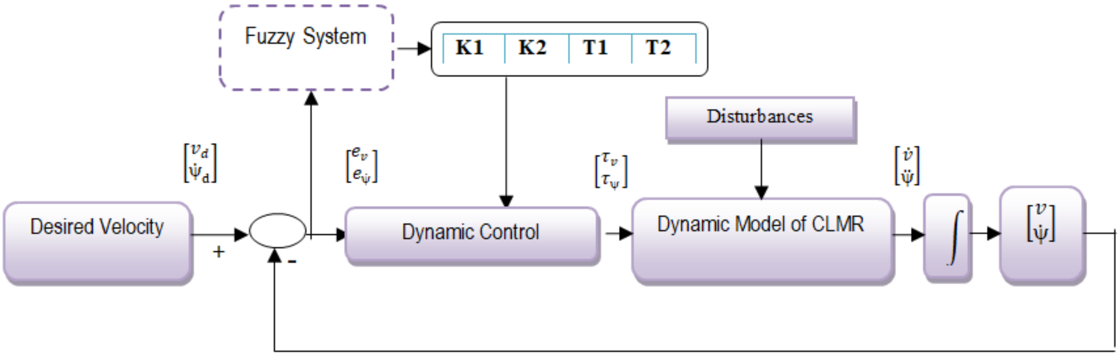
Fuzzy System
K1
K2
T1
T2
Disturbances
$\begin{bmatrix} v_d \\ \dot{\psi}_d \end{bmatrix}$
$\begin{bmatrix} e_v \\ e_\psi \end{bmatrix}$
$\begin{bmatrix} \tau_v \\ \tau_\psi \end{bmatrix}$
$\begin{bmatrix} \dot{v} \\ \ddot{\psi} \end{bmatrix}$
Desired Velocity
+
-
Dynamic Control
Dynamic Model of CLMR
$\int$
$\begin{bmatrix} v \\ \dot{\psi} \end{bmatrix}$

Fig. 7. The system architecture of the closed-loop system by fuzzy logic

Thus the parameters are gauged by the following fuzzy inference mechanism:

- If $e_v$is N and if $e_\psi$is N then k_1 is S, $and\ k_2$,is S and T_1 is S and T_2 is S.
- If $e_v$is N and if $e_\psi$is Z then k_1 is S, $and\ k_2$,is M and T_1 is S and T_2 is M.
- If $e_v$is N and if $e_\psi$is P then k_1 is S, $and\ k_2$,is B and T_1 is S and T_2 is B.
- If $e_v$is Z and if $e_\psi$is N then k_1 is M, $and\ k_2$,is S and T_1 is M and T_2 is S.
- If $e_v$is Z and if $e_\psi$is Z then k_1 is M, $and\ k_2$,is M and T_1 is M and T_2 is M.
- If $e_v$is Z and if $e_\psi$is P then k_1 is M, $and\ k_2$,is B and T_1 is M and T_2 is B.
- If $e_v$is P and if $e_\psi$is N then k_1 is B, $and\ k_2$,is S and T_1 is B and T_2 is S.
- If $e_v$is P and if $e_\psi$is Z then k_1 is B, $and\ k_2$,is M and T_1 is B and T_2 is M.
- If $e_v$is P and if $e_\psi$is P then k_1 is B, $and\ k_2$,is B and T_1 is B and T_2 is B.

6. SIMULATION RESULTS

A simulation study has been conducted to assess the effectiveness of the proposed control. The SC performances are evaluated. The control objective is to make the linear and steering velocities converge to the desired references. To investigate the effectiveness of the proposed methodologies, numerical simulations were carried out for the SC without disturbance. To test the robustness of the approaches, disturbances were added in the second scenario of the simulation.

Let us consider:

- The desired linear velocity: $V_d = 1.0 m.s^{-1}$ and the desired steering velocity: $\dot{\psi}_d = 1.0\ rad.s^{-1}$.

The mobile robot parameters are given in Tab. 1.

Tab. 1. Parameters used in the simulation

Mobile Robot			
$I_\theta(Kgm^2)$	$I_w(Kg\_m^2)$	m (kg)	d(m)
1.35	5*10-3	25	0.2

This part presents the results of simulations of SC to judge the performances achieved by the manual gain, GWO algorithm and fuzzy logic type 2. We have used some metrics for this purpose to evaluate the mean linear velocity and the mean steering velocity errors. The tracking of the linear and steering velocities with the best gains are depicted in Figs. (8) and (9). The controller is tuned to have the fastest response. In these figures, one can figure out how fast is the response of the mobile robot to reach the reference velocities. To highlight this aptitude, trajectories errors are added in Figs. (10) and (11) while figures (12) and (13) present the control torques (14) and (15) present the macro-variable function.

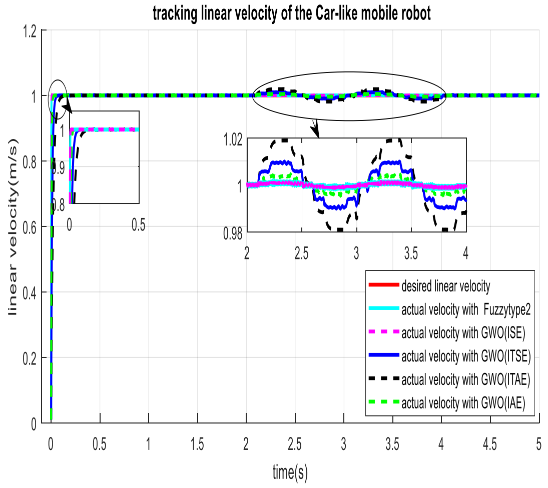

Fig. 8. Linear velocity

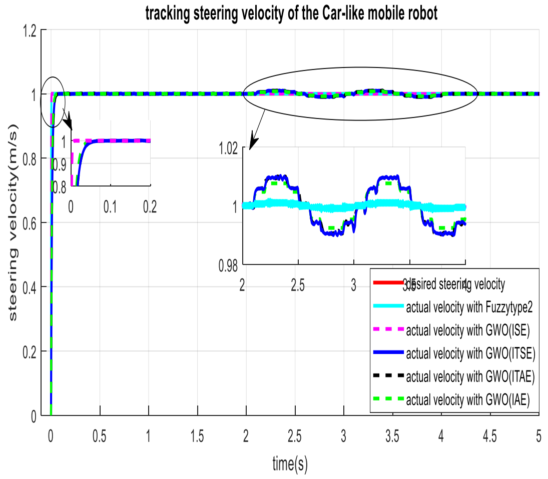

Fig. 9. Steering velocity

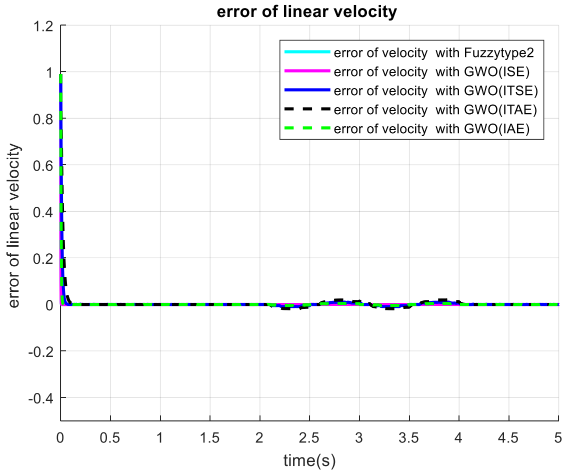

Fig. 10. Tracking error of linear velocity

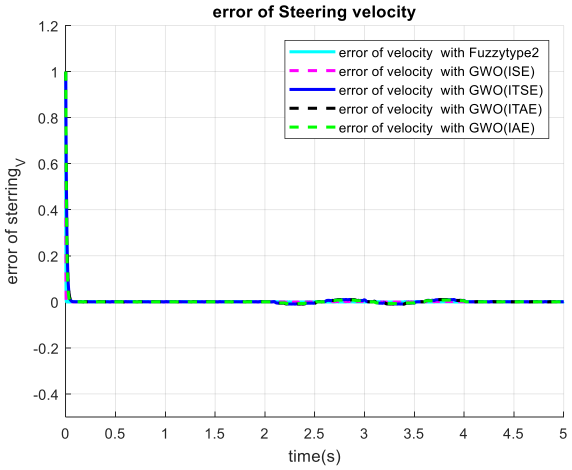

Fig. 11. Tracking error of steering velocity

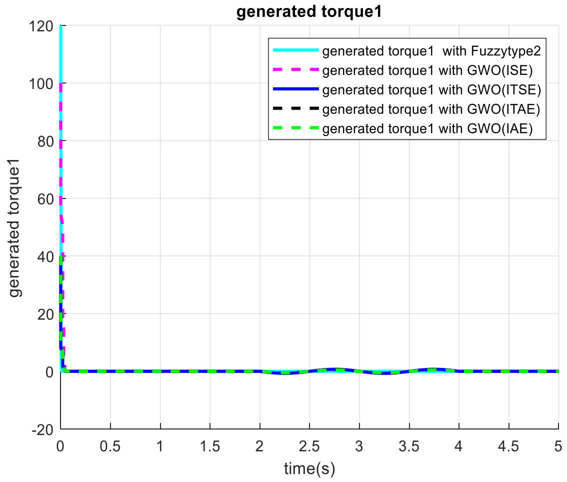

Fig. 12. Generated torque1

We should point out that the SC law used the best optimal parameters found by the GWO algorithm and fuzzy logic type2.

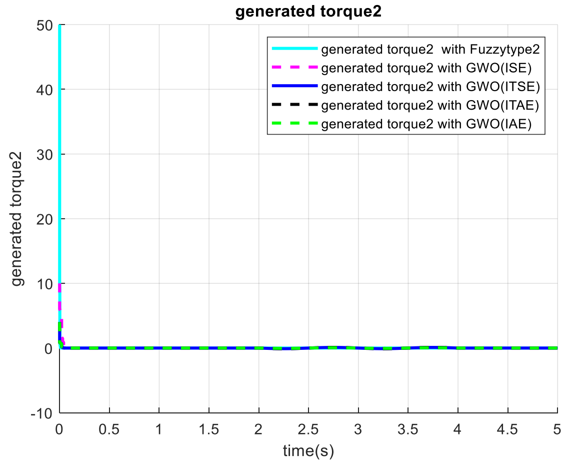

Fig. 12. generated torque2

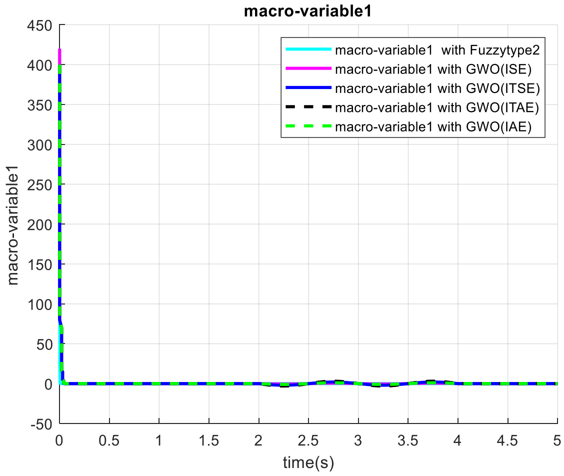

Fig. 14. Macro-variable1

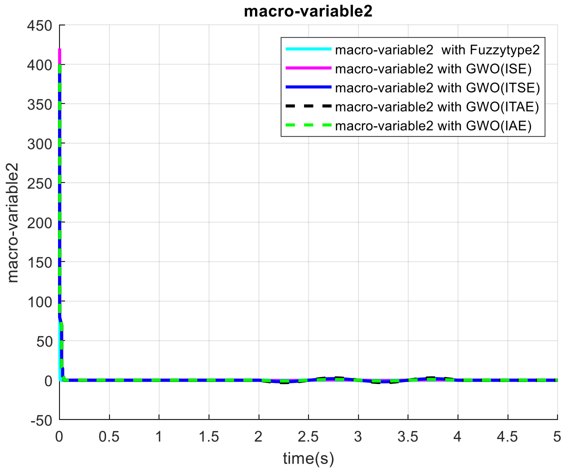

Fig. 15. Macro-variable2

In fact, after execution of the algorithm for 25 generations, using the hyper-parameters of the GWO given in Table 2, we report in Table 3 the resulting optimum values of the gains and the rate of convergences for two chosen objective functions given by expressions (31), (32), (33) and (34). As for thesake of comparison, we start our simulation by trial and error on the unknown parameters. We tried to fix them manually and look for the best performances as it is depicted in Tab. 2. For each set of the parameters: k_1, k_2, $T_1$and T_2, we compare the different metrics defined in the first column of Tab. 3. As one can notice, the time taken for the robot to start tracking the desired velocities is within 0.5 swith a convergence time of about 3 s. The rising time of the steering velocity is decreased when its rate of convergence is reduced and the value of the gain is increased. However, the time of convergence and the mean errors of velocities become small as well as the mean error of the steering velocity. From Tab. 3, we can see that the best parameters are those given in column three, compromised by the time of convergence. We notice also that the time taken for the robot to start tracking the desired velocities is very fast with a convergence time of almost 3 s. The rising time of the steering velocity can decrease when its rate of convergence is reduced and the gain is increased. However, one can remark that the values plugged in this table were very hard to find by trial and error. Therefore, we cannot assert that these parameters are the best ones to span the whole parameter space and choose the values that can give the best performances automatically. In this case, we ran the GWO algorithm and recorded the performances found and plugged them in Table II. To observe the effect of the parameter values, a comparison between the results obtained using manual trials, the GWO algorithm and those of Fuzzy logic type 2. As it can be seen, that the Fuzzy logic type 2 and the value of GWO obtained from the integral of the error absolute achieved the best records.

Moreover, to test the robustness of each of the approaches, we inject some disturbances into the system. By trial error, we adjust manually the parameters until we observe that the velocities approach as close as possible their references. Despite the optimum parameters found by the integral of the square of the time-weighted error TISE of the GWO algorithm, it remains to enable to eliminate residual disturbances. To arrive at a satisfactory objective. The SC based on fuzzy logic type 2 parameter finding has solved the problem. In fact, the ability to find the appropriate parameter values at each sampling time helped in reducing enormously the effect of disturbances, where, one can observe an excellent response of tracking the forward and steering velocities.

It can be seen that the system operates on the manifold $\psi = 0$. The convergence rate of the dynamical system is determined by the value of the parameter *T*, the smaller the *T* value, the higher the convergence rate, the figures of simulation demonstrate that the convergence rate can be controlled.

Tab. 2. GWO parameters of synergetic

Objective function	IAE	ITAE	ISE	TISE
Gain	k_1 = 480 k_2 = 480	k_1 = 401 k_2 = 500	k_1 = 455 k_2 = 350	k_1 = 455 k_2 = 497

The rate of convergence	T1 = 0.011
	T2 = 0.00
		T1 = 0.01
		T2 = 0.0
			T1 = 0.02
			T2 = 0.01
				T1 = 0.004
				T2 = 0.008

Tab. 3. Mean error of synergetic control

	Manual gain			**GWO**				**Fuzzy logic Type 2**
Gain	$k_1 = 10$ $k_2 = 10$	$k_1 = 500$ $k_2 = 500$	$k_1 = 500$ $k_2 = 500$	$k_1 = 480$ $k_2 = 480$	$k_1 = 401$ $k_2 = 500$	$k_1 = 451$ $k_2 = 350$	$k_1 = 455$ $k_2 = 497$	
The rate of convergence	$T_1 = 0.1$ $T_2 = 0.1$	$T_1 = 0.1$ $T_2 = 0.1$	$T_1 = 0.01$ $T_2 = 0.01$	$T_1 = 0.011$ $T_2 = 0.001$	$T_1 = 0.01$ $T_2 = 0.01$	$T_1 = 0.02$ $T_2 = 0.01$	$T_1 = 0.004$ $T_2 = 0.003$	
Mean linear velocity error (m/s)	0.0646	0.0561	0.0123	0.0048	0.0014	0.0194	0.0042	0.002
Mean steering velocity error (m/s)	0.0654	0.0561	0.0120	0.0014	0.0049	0.0131	0.0036	0.002
Time of convergence (s)	3	3	3	9	9	3	7	30
Rising time of linear velocity (s)	0.5	0.5	0.08	0.01	0.05	0.2	0.03	0.01
Rising time of steering velocity(s)	0.5	0.5	0.08	0.08	0.06	0.1	0.02	0.02

7. CONCLUSION

In this paper, an optimised SC is proposed. The SC is designed to track the forward and steering velocities of a CLMR. The control law obtained involved parameters that should be well tuned to get the best performances. The tuning has been performed using the GWO optimisation algorithm and fuzzy logic type 2. The simulation confirmed a rapid convergence of the control law. Moreover, applying this control law helps in avoiding the chattering problem while assuring asymptotical stability. To arrive at the best performances, we used the GWO algorithm and fuzzy logic type 2 to get the optimised parameters of the control law. The SC law based on fuzzy adaptive parameters can choose the appropriate values of the parameters to best remove the external disturbances, which demonstrates its robustness. The results obtained are very satisfactory and promise a lot in the use of this control law in driving an autonomous vehicle in real-time conditions.

These control laws of linear and steering velocity can assure the asymptotical stability of the system by applying the Lyapunov theory, and proves that the controller is stable for any combination of the error states. The advantage of this control law is to eliminate the disturbances due to the dynamic model, such that the error states of the robot converge to zero.

REFERENCES

1. Ahifar A, Ranjbar AN, Rahmani Z. Finite Time Terminal Synergetic Controller for Nonlinear Helicopter Model. 2019; 32(2):236–241.
2. Benaziza W, Slimane N, Mallem A. Disturbances elimination with fuzzy sliding mode control for mobile robot trajectory tracking. Advances in Electrical and Electronic Engineering. 2018; 16(3):297–310. https://doi.org/10.15598/aeee.v16i3.2767
3. Bhattacharyya S, Shimoda S, Hayashibe M. A Synergetic Brain-Machine Interfacing Paradigm for Multi-DOF Robot Control. IEEE Transactions on Systems Man and Cybernetics: Systems 2016; 46(7):957–968. https://doi.org/10.1109/TSMC.2016.2560532
4. Dung NM, Duy VH, Phuong NT, Kim SB, Oh MS. Two-wheeled welding mobile robot for tracking a smooth curved welding path using adaptive sliding-mode control technique. International Journal of Control Automation and Systems. 2007; 5(3):283–294.
5. Elhariri E, El-Bendary N, Hassanien AE, Abraham A. Grey wolf optimization for one-against-one multi-class support vector machines. 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR). 2015; 7–12. https://doi.org/10.1109/SOCPAR.2015.7492781
6. Gupta S, Deep K. Cauchy Grey Wolf Optimiser for continuous optimisation problems. Journal of Experimental & Theoretical Artificial Intelligence. 2018; 30(6):1051–1075 https://doi.org/10.1080/0952813X.2018.1513080.
7. Humaidi AJ, Ibraheem IK, Azar AT, Sadiq ME. A New Adaptive Synergetic Control Design for Single Link Robot Arm Actuated by Pneumatic Muscles. Entropy. 2020; 22(7). https://doi.org/10.3390/e22070723
8. Ibrahim AEB. Wheeled Mobile Robot Trajectory Tracking using Sliding Mode Control. 2016. https://doi.org/10.3844/jcssp.2016.48.55
9. Kamalova A, Navruzov S, Qian D, Lee SG. Multi-Robot Exploration Based on Multi-Objective Grey Wolf Optimizer. Applied Sciences. 2019; 9(14). https://doi.org/10.3390/app9142931
10. Kolesnikov A, Veselov G, Kolesnikov A. Modern applied control theory: synergetic approach in control theory. TRTU Moscow Taganrog. 2000; 4477–4479.
11. Liu CH, Hsiao MY. A finite time synergetic control scheme for robot manipulators. Computers and Mathematics with Applications. 2012; 64(5):1163–1169. https://doi.org/10.1016/j.camwa.2012.03.058
12. Mallem A, Slimane N, Benaziza W. Dynamic Control of Mobile Robot Using RBF Global Fast Sliding mode. IAES International Journal of Robotics and Automation (IJRA). 2018; 7(3):159. https://doi.org/10.11591/ijra.v7i3.pp159-168
13. Mesquita EDEM, Sampaio RC, Vicente H, Ayala H, Llanos CH. Recent Meta-Heuristics Improved by Self-Adaptation Applied to Nonlinear Model-Based Predictive Control. 2020; 118841–118852.
14. Mirjalili S, Mirjalili SM, Lewis A. Grey Wolf Optimizer. Advances in Engineering Software. 2014; 69:46–61. https://doi.org/https://doi.org/10.1016/j.advengsoft.2013.12.007
15. Mittal N, Singh U, Sohi BS. Modified Grey Wolf Optimizer for Global Engineering Optimization. Applied Computational Intelligence and Soft Computing. 2016; 1–16. https://doi.org/10.1155/2016/7950348

16. Peng S, Shi W. Adaptive fuzzy integral terminal sliding mode control of a nonholonomic wheeled mobile robot. Mathematical Problems in Engineering. 2017. https://doi.org/10.1155/2017/3671846
17. Podvalny SL, Vasiljev EM. Synergetic control of UAV on the basis of multi-alternative principles. International Russian Automation Conference RusAutoCon. 2018; 1–6. https://doi.org/10.1109/RUSAUTOCON.2018.8501727
18. Sklyarov AA, Veselov GE, Sklyarov SA, Pohilina TE. Synthesis of the synergetic control law of the transport robotic platform. Proceedings of 2017 IEEE 2nd International Conference on Control in Technical Systems CTS. 2017; 285–288. https://doi.org/10.1109/CTSYS.2017.8109547
19. Veselov G, Sklyrov A, Mushenko A, Sklyrov S. Synergetic Control of a Mobile Robot Group. Proceedings - 2nd International Conference on Artificial Intelligence Modelling and Simulation AIMS. 2014; 155–160. https://doi.org/10.1109/AIMS.2014.22
20. Yeh YC, Li THS., Chen CY. Adaptive fuzzy sliding-mode control of dynamic model based car-like mobile robot. International Journal of Fuzzy Systems. 2009; 11(4):272–286. https://doi.org/10.30000/IJFS.200912.0006.

Zoulikha Bouhamatou: https://orcid.org/0000-0002-3985-0147

Foudil Abedssemed: https://orcid.org/0000-0003-0935-3147